Subject: Re: MacOS X woes
Posted by lundman on Sat, 16 Dec 2006 01:30:57 GMT
View Forum Message <> Reply to Message

When I rsynced the new Util.h it over-wrote my port changing, including setting of flagPPC. Sigh.

Anyway, back to where we were, looking at the image editor segv.

Called LoadIml(), which does not find IMAGE_ID and IMAGE_DATA, so it throws the exception, and loads in a more low-level loader.

Seems to parse the input ok, until is reaches:

```
(gdb) p parser
$28 = (CParser &) @0xbffe5a70: {
  term = 0x2161732
"IMAGE_SCAN(\"?\\377\\377\\377\\13?????????????????????????????\\377\\377\\377\")\nI
MAGE_PACKED(ufxp,
\"\\2\\20\\0\\0\\0\\20\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\")\nIMAGE_BEGIN(options)\n\tIMAGE_
SCAN(\"?\\1vlV????\\1??m?\\2??s?"...,

100                         else if(id == "IMAGE_PACKED" && parser.IsChar('\"'))
(gdb) n
105                                 StringStream ss(d);
(gdb) n
106                                 ss % image;

(gdb) n
107                                 if(!ss.IsError())
(gdb) n
108                                     accepted = true;
(gdb) n
109                             }
(gdb) n
112                     if(name.GetLength() >= 6 && !memcmp(name, "_java_", 6))
(gdb) p name
[snip]
    ptr = 0x1b89654 "ufxp"
(gdb) n
115                     if(accepted)
(gdb) n
117                         if(name.GetLength() >= 4 && !memcmp(name, "im__", 4))
(gdb) n
120                         Image m = RLEToAlpha(encoded_data, image.size);
(gdb) p encoded_data
$34 = {
```

```
  <AString<char,String>> = {
    <Moveable<String,EmptyClass>> = {
      <> = {<No data fields>}, <No data fields>},
    members of AString<char,String>:
    ptr = 0x20c9414
"????\001??????????????????????\004??????????????????\002??????????????\004???[[[???
??????\003ooo?????????????\n???\036\036\036\033\033\033??????ooo\f\f\f\016\016\016???
???????????\n999\001\001\001"
  }, <No data fields>}
(gdb) p image.size
$35 = {
  <Moveable<Size_<int>,EmptyClass>> = {
    <> = {<No data fields>}, <No data fields>},
  members of Size_<int>:
  cx = 268435456,
  cy = 268435456
}
```

Should sizes be set here? Before we call, or is it just uninitialised?

encoded_data looks correct, in that it was parsed in ok.

```
(gdb) x/16bx encoded_data.ptr
0x20c9414:      0x83    0xff    0xff    0xff    0x01    0xfd    0xfd    0xfd
0x20c941c:      0x82    0xfc    0xfc    0xfc    0x83    0xff    0xff    0xff
```

But it dies in RLEToAlpha.

```
Breakpoint 3, RLEToAlpha (rle=@0xbffe5898, sz=@0xbffe5850) at ImlFile.cpp:32
32              ImageBuffer ib(sz);
(gdb) p sz
$39 = (Size &) @0xbffe5850: {
  <Moveable<Size_<int>,EmptyClass>> = {
    <> = {<No data fields>}, <No data fields>},
  members of Size_<int>:
  cx = 268435456,
  cy = 268435456
```

Makes me think the size should be "somewhat smaller".

You set the size at IMAGE_END phase, I don't recall it reading IMAGE_END. Could it be our .iml

file is incorrect,old ?

Has IMAGE_BEGIN(ufxp)
 IMAGE_SCAN x 16
IMAGE_PACKED(ufxp, ...)
Then next IMAGE_BEGIN. etc. (of two)